Cryptography - A Crash Overview

Stanisław Radziszowski
Rochester Institute of Technology
spr@cs.rit.edu

Gdańsk, November 2010
Rochester, January 2015
Henrietta, August 2022

Cryptography
goals

Desired security properties in the digital world:

- confidentiality, secrecy
- data integrity
- authentication, of data origin and entity
- non-repudiation

Cryptography

limits

Cryptography is an important, but only
a relatively small part of security:

- right choice of tools is hard
- implementation errors are common
- variety of side-channel attacks can bypass best crypto
- social attacks

Cryptography

primitives and algorithms

Primitives, algorithms and protocols often are classified into three categories:

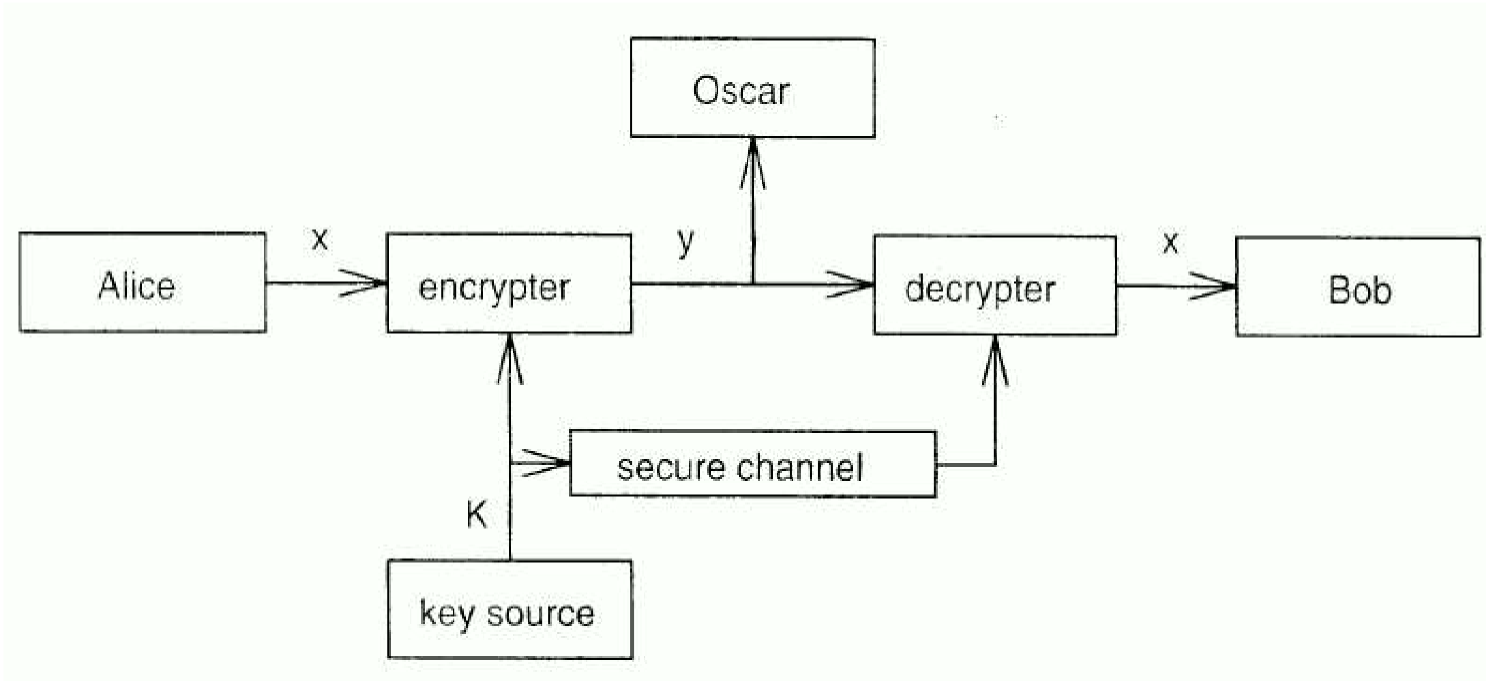
- unkeyed: no keys, checksumming, fingerprinting, hashing
- symmetric-key: shared key, often called private-key
- asymmetric-key: no shared key, called public-key,

  has private/public keys, often much more and a growing complex public-key infrastructure (PKI)

R · I · T
Department of Computer Science

Cryptography

basic symmetric-key scenario



[Stinson]

Cryptography

unkeyed primitives and algorithms

Unkeyed practice and theory

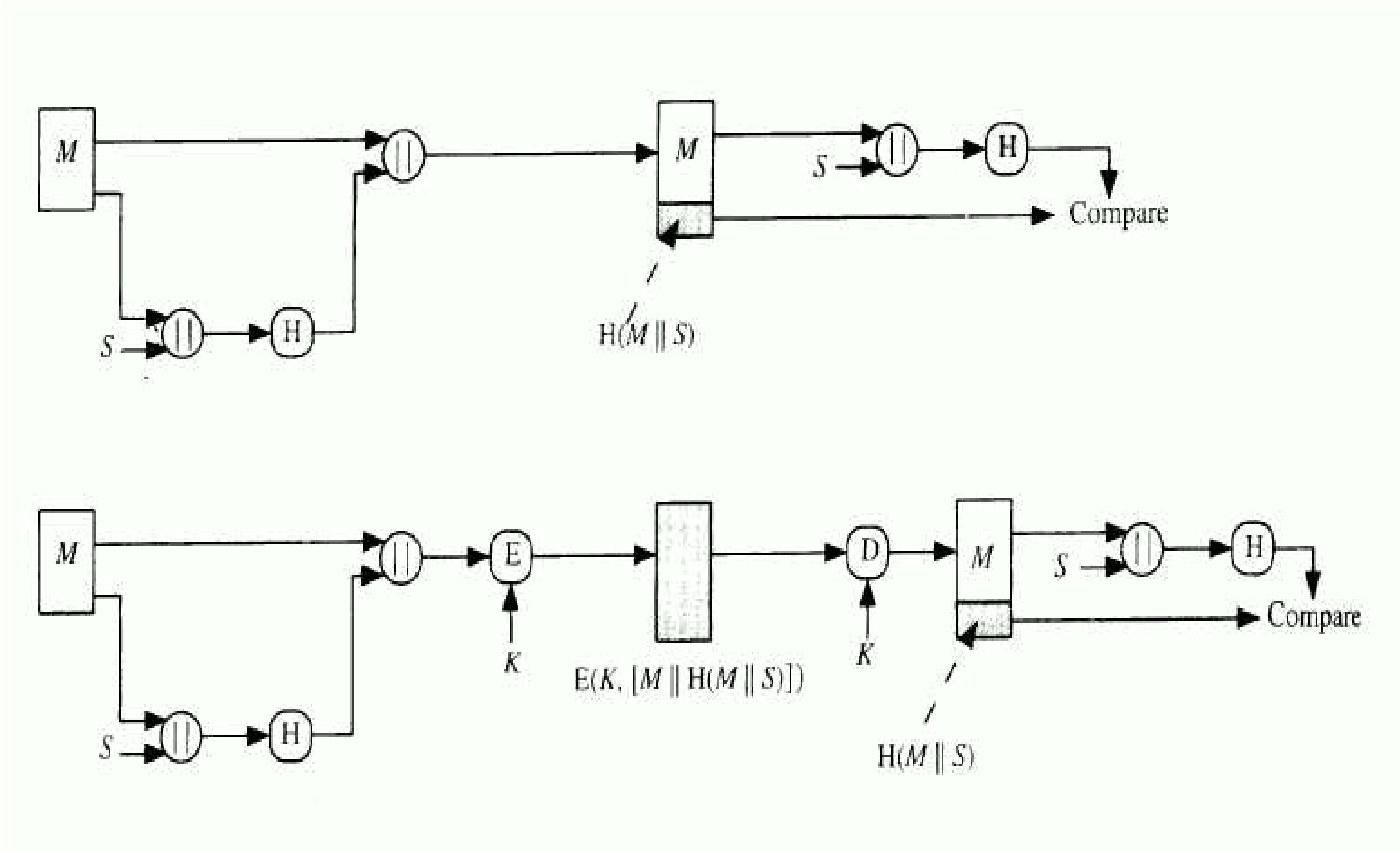
- hashing, SHA-family, (P)RNG
- one-way permutations exist, or **NP** is not that much ...

Use

- key generation
- hash and sign paradigm
- pseudo-random sequences - Blum-Blum-Shub BBS generator, stream cipher outputs, PRNGs

Hash in Use

message authentication - clear and encrypted



[Stallings]

SHA-3 = Keccak 2007 – 2050

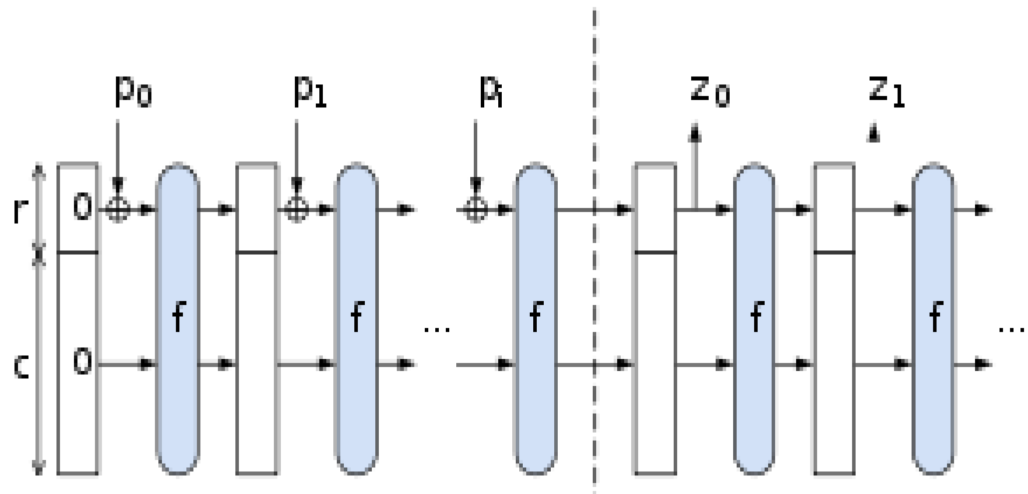
the main hash you may end up using

- Winner of the 2007-2012+ NIST SHA-3 competition
  Draft FIPS 202, May 2014
- Team (from STMicroelectronics and NXP Semiconductors):
  Guido Bertoni, Joan Daemen (of the AES fame),
  Michaël Peeters, Gilles Van Assche
- hashing SHA3-224, SHA3-256, SHA3-384, SHA3-512
- extendable-output function SHAKE128, SHAKE256

- Elegant, convincing sponge design, ideas from *Grindahl*
- Runs on a $5 \times 5 \times 2^l$ cube of bits,
  recommended 1600-bit state ($l = 6$)

SHA-3/SHAKE/Keccak sponge

r absorption *rate*
c security *capacity*
f crypto reshuffling *permutation*
p absorbed *message*
z squeezed *output*



[wikipedia]

Cryptography

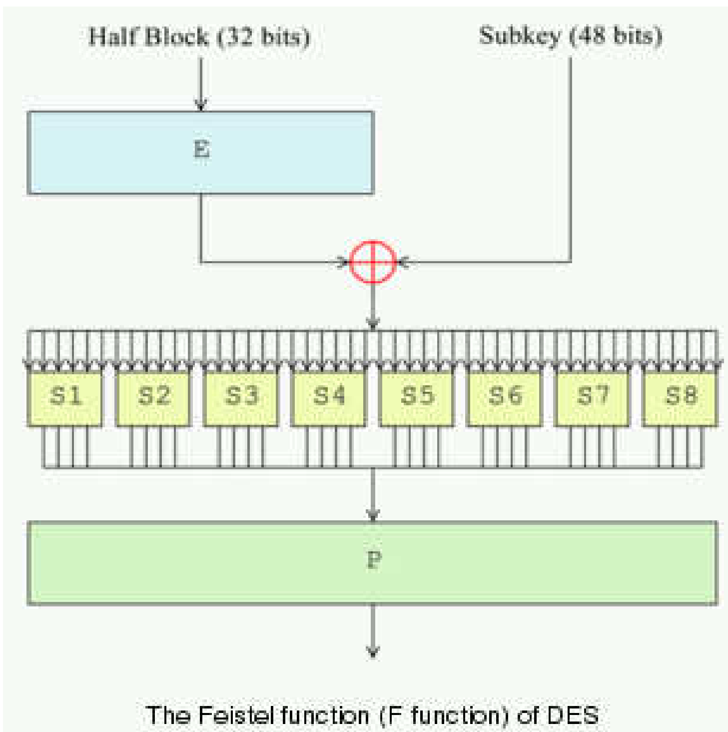
shared-key

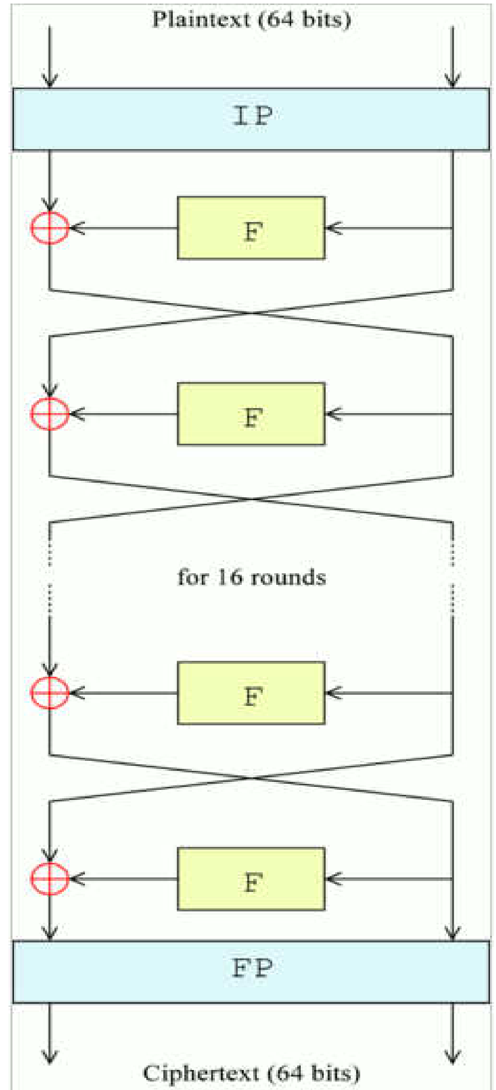
Symmetric keys

- block ciphers since 1970's
  IBM's Lucifer,
  DES - Data Encryption Standard, 3DES
  IDEA - International Data Encryption Algorithm
  AES - Advanced Encryption Standard
- stream ciphers, troubled RC4
- can be based on CTR mode of block ciphers
- MAC, HMAC - message authentication codes
- PRNG - pseudo-random number generators

Data Encryption Standard (1977 – 1998 – ...?)

Feistel cipher



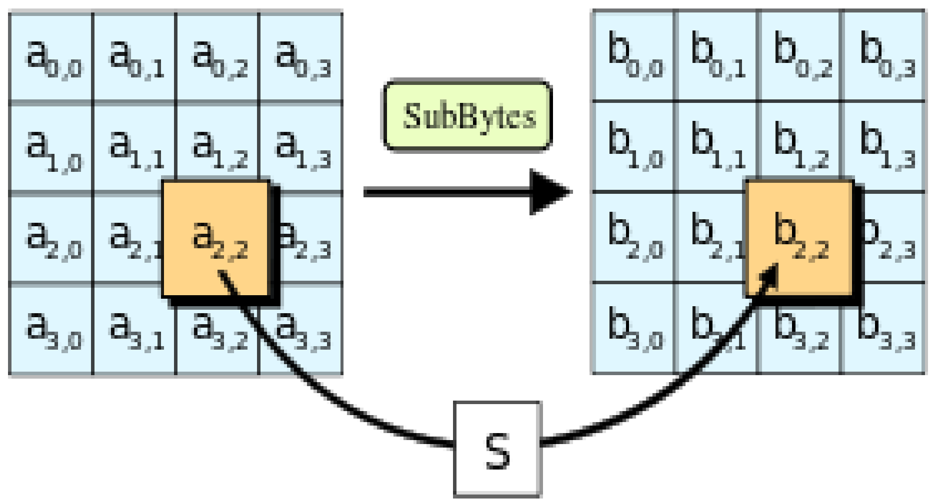


[Wikipedia]

Advanced Encryption Standard (1997 – 2001 – ...?)

all steps invertible in Galois fields

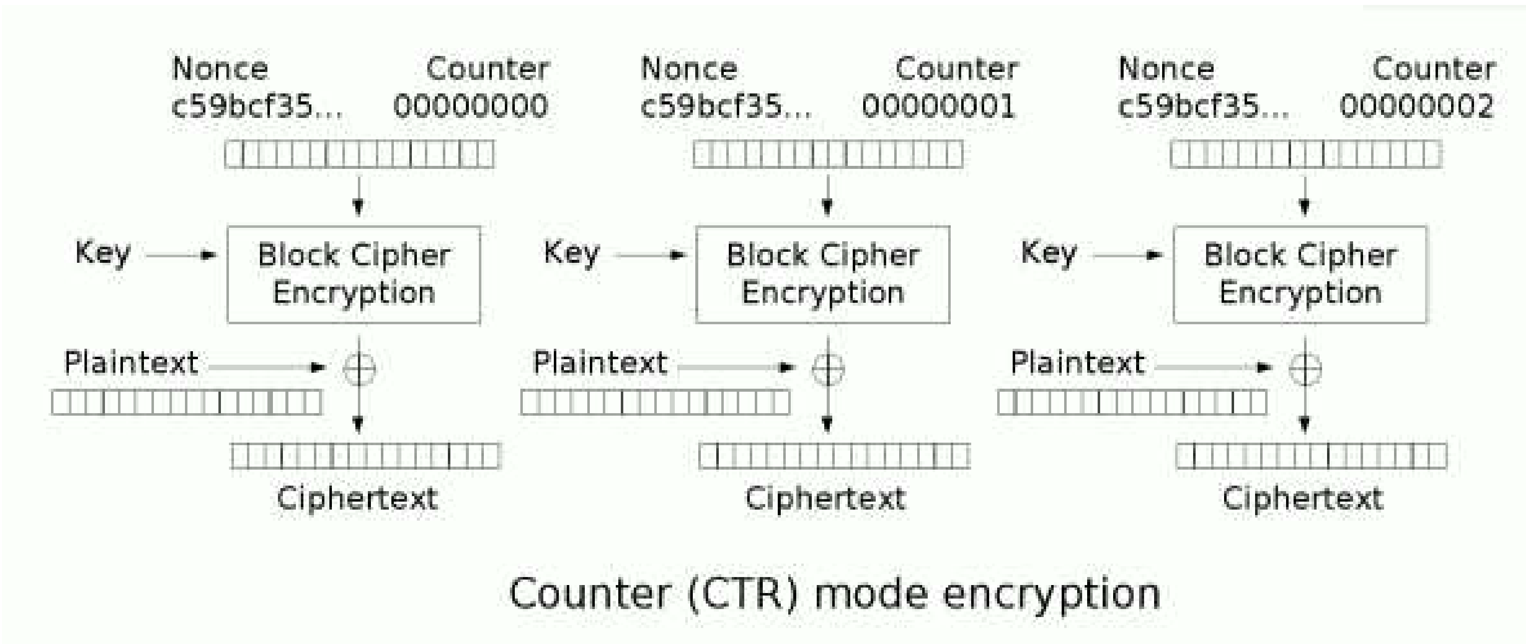
- Rijndael by Vincent Rijmen and Joan Daemen, BE
- winner of the NIST cipher 1997 – 2001 competition
- state is a 4 $\times$ 4 matrix of bytes in $GF(2^8)$



[Wikipedia]

Block Cipher in Use

shared key



[Wikipedia]

Cryptography

public-key primitives and algorithms

Public keys:

- Public-key cryptosystems
  RSA - Rivest, Shamir, Adleman
  ElGamal, McEliece cryptosystems
  ECC - elliptic curve cryptosystems
- Signatures
  DSS/DSA - Digital Signature Standard/Algorithm
  ECDSA - Elliptic Curve Digital Signature Algorithm
- PKI - public-key infrastructure, only if we had it right :-(
  DH - Diffie-Hellman key agreement
  key management, distribution and X.509
- Homomorphic cryptography - Paillier, Gentry, BGV
- Post-quantum cryptography

Main Public-Key Systems in Use

RSA and ECC

RSA by Rivest-Shamir-Adleman, 1977
has an edge over ECC, because

- it is simple and well understood
- links nicely to basic number theory
- deployed earlier on many systems

ECC by Koblitz-Miller, 1985
has an edge over RSA, because

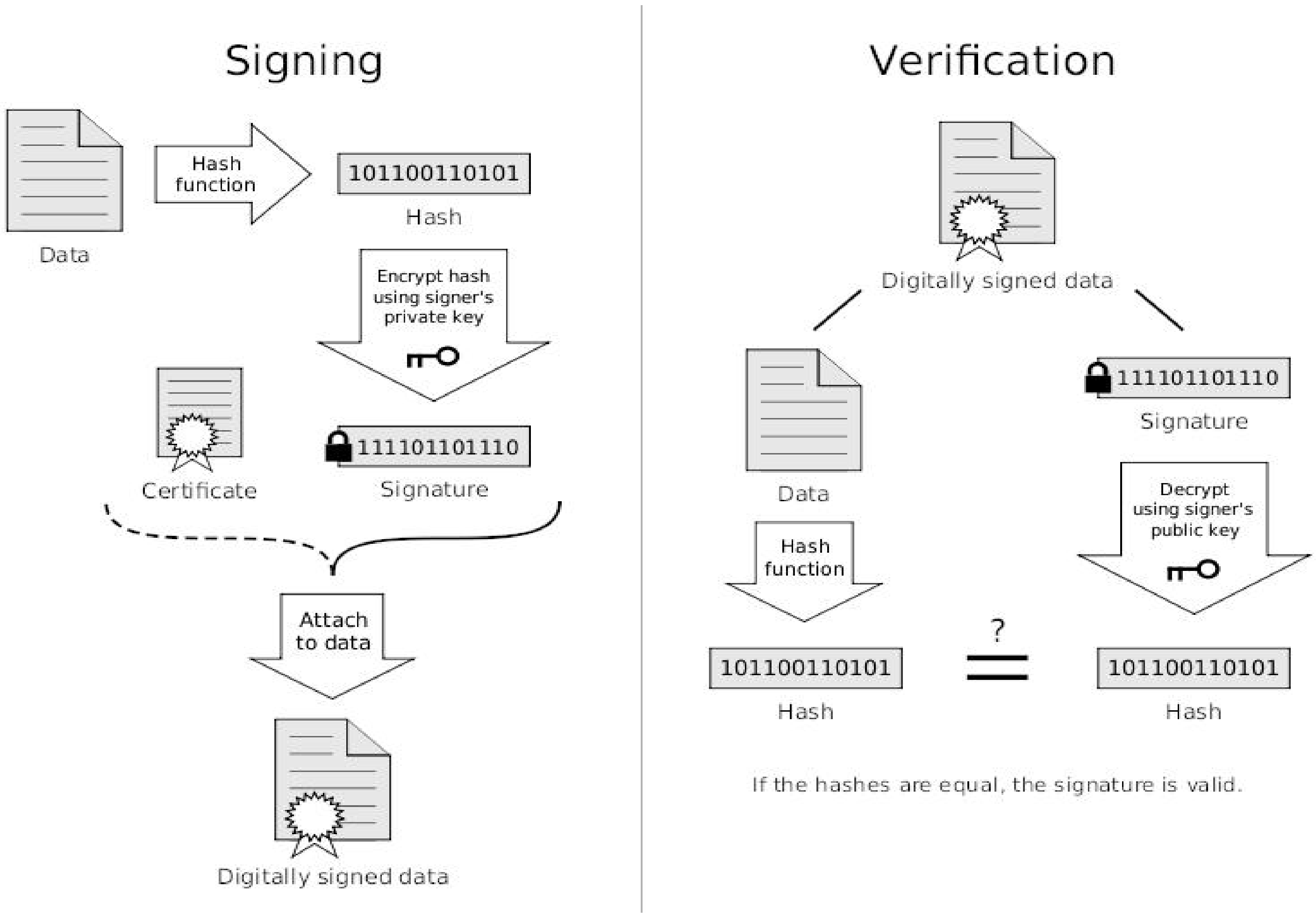
- it uses short keys (163+ bits ECC vs. 1024+ bits RSA)
- delivers much better performance
- ECC uses great theory of elliptic curves on top of classical number theory used by RSA

Prediction: ECC will take over, unless QC and PQC win

R · I · T
Department of Computer Science

Public-key System in Use

signature by hash and public-key encryption



[Wikipedia]

Special Functionalities

using proofs and money

- Zero-knowledge protocols
- Authenticated encryption
  CAESAR competition: 2012 – 2017/2018
- Electronic voting: no individual vote audit
- Oblivious transfer, two millionaires problem
- Electronic cash: untraceable, no double-spending, bank-shop-customer roles, central bank
- Cryptocurrencies, SHA or scrypt POW based:
  Bitcoin BTC, fully distributed, anonymity questioned,
  Ethereum ETH, and many more ...

Quanta

QC, QKD and PQC

- QC - quantum computing :-(:-(
- QKD - quantum key distribution :-(
- PQC - post-quantum cryptography :-)

  NIST competition 2017-2023 is finishing:

  CRYSTALS-KYBER for key-establishment, and
  CRYSTALS-Dilithium for signatures

  will be standardized

Mathematics in Cryptography

Math in primitives

- Keyless: up to SHA-2 mostly bit juggling, much more is in SHA-3
- Shared-key: much more since AES ’2001, much based on binary Galois fields $GF(2^k)$
- Public-key: heavy use of number theory, now in most schemes, including ECC. But lattices are coming ...

Math in cryptanalysis

- Linear and differential cryptanalysis
- Probability and statistics, random oracle models
- Number theoretical algorithms: primality, factoring
- Discrete logarithms: cyclic group discovery, index calculus, counting points on elliptic curves, theory of elliptic curves

R·I·T
Department of Computer Science

Cryptography Engineering
evaluation criteria

Security engineer must consider:

- **Level of security.** Or, how many security bits you need.
- **Functionality.** Or, how primitive are the primitives.
- **Performance.** Or, how fast is fast enough.
- **Simplicity.** Is there still anybody who can understand it?

Each party stresses a different measure:

- risk (politicians)
- cost (managers)
- use (most of us)

Can security/software engineer satisfy all of them?

References

- Niels Ferguson, Bruce Schneier and Tadayoshi Kohno, *Cryptography Engineering*, John Wiley & Sons, 2010.
- Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, *CRC Handbook of Applied Cryptography*, CRC Press 1996
  `http://www.cacr.math.uwaterloo.ca/hac`
- Bruce Schneier, *Crypto-Gram Newsletter*, current,
  `https://www.schneier.com/crypto-gram`
- William Stallings, *Cryptography and Network Security. Principles and Practice*, 7-th edition, Prentice Hall, 2018.
- Douglas R. Stinson and Maura B. Paterson, *Cryptography: Theory and Practice*, 4-th edition, CRC Press 2019.